Inter Office Memo

ATARI®

Coin Operated Games Division

To: Distribution

From: Jack Ritter

Subject: Project Initiation Meeting - OFF-ROAD Date: 2/5/85

You are invited to a project initiation meeting for OFF-ROAD on:

Friday
February 8, 1985
11:00 am
Lyle's Conference Room

JR:yt

Distribution:

John Bell
Mary Fujihara
Brad Fuller
Rick Moncrief
Rich Moore
Lyle Rains
Jack Ritter
Dave Stubben
Chuck Swenson
Don Traeger
Dan Van Elderen

Off-Road Game Spec. J. Ritter

****
O V E R V I E W
****

Off-Road is a driving game. It takes place in a desert-like, off-road environment. The surface you drive on is a varying terrain: dunes, mud holes, gravel fields, rivers, etc. The perspective is overhead, and a little behind. The screen is a scrolling window into a much larger universe.

The driving surface varies in slope as well as stability, and so TRACTION and TIRE TORQUE are factors. This means that good use of shifting and accelerating are the key to negotiating the course well. As you drive over dunes, mud flats, etc, your all-purpose, 4-wheel-drive vehicle will do lots of fish-tailing and bucking. Steering is limited so that you can only make forward progress.

At the start of the game, you leave the starting line in last place. The object of the game is to pass as many other vehicles as possible, crossing the finish line in the lowest place possible.

The locations of the other vehicles, and of you, are indicated by a global map. The map is a thin vertical rectangle, which represents locations as specially colored dots. This global map is therefore simple to read, and takes up very little screen area.

There are no lateral boundaries, as in other driving games. You may sweep to the left or right as desired, to avoid obstacles, or to drive to a more desirable surface. The terrain is generated dynamically ahead of you. There are no left or right scrolling limits.

As you catch up to a vehicle, it will automatically drift left or right as necessary, so that it will be on the screen as you pass it. This means that you can't "sneak" past a vehicle. Each passing of a rival vehicle is a confrontation. Sometimes you will loose speed, and drop DOWN in place; i.e, a previously passed vehicle will overtake you. As it catches up to you, it again will drift as needed so that it is in view during the passing confrontation.

Distributed on the terrain are various obstacles: rocks, cacti, etc. If you hit any of these, they will jolt you in various ways, temporarily hindering your forward progress.

There are also moving obstacles (other than the other vehicles in the race): tumble weeds, dirt bikers, mules, crazed prospectors, Indian spirits, galloping skeletal horses, etc. These moving things will tend to move laterally into your path, and should be avoided. They are incidental characters which are not part of the race, but exist to liven things up. (More on tumble weeds in the Obstacles Section).

Also distributed on the terrain are GOOD items: extra tires and nitro fuel cannisters. Driving over a good thing snaps it up, for immediate, or subsequent use. The locations of goodies are shown on the global map.

The other vehicles you are racing have individual personalities. They vary in their ability to negotiate the different types of terrain. Some are

good in the mud but slower on open sand, etc. Some kick up mud, some are jet powered and throw back a dangerous jet of exhaust, some drop obstacles in your path, etc. Even though they have customized features, they all must "obey" the terrain, as you must, although they are bound by the rules of traction and grade in varying ways.

As you pass a vehicle, you may bump into it. If you hit it too hard, you will flip over a few times and land upright, facing forward, with no speed. You then simply tear off again. This flip/stop sequence is therefore a time penalty, and not a death. All collisions, whether with terrain obstacles or other vehicles, are time penalties, via temporary loss of control and/or speed. There is no crashing and dying.

The maximum speed achievable is about half that of Spy Hunter, in terms of screens per second of background. This is so there is time to interact with the terrain. One half the speed of conventional pavement speeds is still pretty fast, and allows the extra dimension of lateral, curving motion.

SCREEN LAYOUT

See Screen 1. The screen is mounted vertically. The global map is the full length of the screen, for maximum resolution in the forward direction. Note the chain of dots in the map; these are the other cars. The length of this chain relative to the length of the entire race (from starting line to finish line) is revealed by the map. This gives the player an idea of where he stands at any given time. It also shows him the general progress of the other vehicles, and a feel for the overall scale of the race. The map also shows the locations of Nitro Fuel Cannisters and tires, which he can navigate to and pick up. All of these location markers will wrap around laterally if the player drives far enough to the left or right. The map does not show the locations of ANY terrain features or obstacles, since these are generated dynamically.

Along the bottom of the screen is an area which displays:

1) Time remaining. This is an analog stripe, which shrinks in length, giving time remaining as a quantity easily read out of the corner of the player's eye.
2) Score: A string of 6 2X2 digits.
3) level: A single 2X2 level number.
4) Nitro Fuel token list: shows Number of nitro fuel units in player's posession.
5) Tire token list: shows number of tires in player's posession. A tire is automatically taken from the list store and put on a worn tire's position when a tire on the player's car gets worn enough.

****
C O N T R O L S
****

In the center is a Spy-Hunter-type steering wheel. Since the player will often have his right hand on the shift stick, the wheel will often be held by his left hand only. The grip type wheel should facilitate this. The wheel turns left or right a maximum of theta (Fig. A), and then is stopped by a mechanical tab. Since theta is less than 90 degrees, it is impossible to circle around and go backwards. Theta determines the maximum rate of lateral progress that the player's path can make. The angle of the player's vehicle won't lock onto the wheel's angle exactly, but will seek toward it, with a lag. A sudden jerk on the wheel will cause the vehicle to fishtail, then right itself to correspond to the angle of the steering wheel.

The shift stick is to the right of the wheel. It is a standard lo-hi, two speed stick.

On the floor and to the right is the accelerator pedal. This is an analog control, as in the usual driving game. Depressing it increases the power output of the engine.

There is one button: the "jet boost" button. Distributed along the terrain are cannisters of NITRO FUEL. You gain one if you drive over it. (Cannister locations are shown on the macro map). If you hit the "jet boost" button (and you have picked up a cannister), you will blast ahead at super speed for one second, flying over ANY vehicle or obstacle ahead of you. As you do so, flames shoot back and a good jet sound is heard.

GAME PROGRESSION

SEE FIG. B. At the start of the race, all the vehicles are lined up on the starting line. Your vehicle is at the right end. The screen/window starts out at the right end, showing you your vehicle, which blinks on and off, distinguishing it from the others (as in Pole Position).

The screen then scrolls LEFT, revealing all the other vehicles in the race. When the screen reaches the left of the line it stops. The start signal is given, and the vehicles tear off straight ahead, IN STAGGERED SUCESSION, FROM LEFT TO RIGHT. As they do so, the screen scrolls to the right, so that you can see each vehicle take off. As the screen approaches your vehicle at the right end, some kind of count down que is shown/heard (like the descending green lite in Pole Position, or something), to que you that your turn is coming. When the screen reaches your vehicle, your controls become active and you tear ahead, last off the line.

This left, then right scrolling serves two purposes:

1) It shows the player that all the vehicles in the race are strung out in a line, and that he is last in that line.

2) It lets the player PREVIEW THE COMPETITION: he can see who is ahead of him, and in what order they are in.

After the other vehicles go, they spread out into a line. The player will soon be able to catch up to the guy immediately ahead of him. There will be N vehicles altogether; where N is 10 - 20. If he keeps up good progress, the average player should be able to overtake about N/3 vehicles. There will be enough time on the clock so that a novice player will have time to get to the finish line, even if he only overtakes a few cars.

The better he does, the more bonus time he is given. A simple table of bonus increments, indexed by finishing rank, will be used. If a player finishes, say, (N/4)th or better, he should have enough time for another full race. Subsequent races increase in difficulty. The level number displayed shows the race in progress. Even a top player will only be able to buy himself a few additional races, so the game is basically fixed-time, as with other driving games. Although the game is ultimately fixed-time, a large point bonus will be awarded whenever a race is finished.

A wide variation in DIFFICULTY can easily be implemented with two parameters:

1) The choice of what mix of vehicles he is pitted against. Some guys are nastier than others (See Fig. C), or, to put it another way, some lineups are worse than others.

2) The sequence of terrain generation (More on this in Terrain Section).

The finish line is a certain DISTANCE from the starting line. If the clock runs out when the player is in the middle of the race, the race, and the game, ends.

****
T E R R A I N
****

The default terrain is regular sand. There can also be dunes, mud patches, gravel patches, or water:

1) REGULAR SAND: Occurs most of the time. Regular sand is actually the default background terrain. The terrain generator overlays the other four terrain types onto sand. (Has high coeff. of static friction).

2) DUNES: A dune is made up of more unstable sand than regular sand (medium coeff. of friction). A dune is the only one of the five terrain types which has non vertical normal vectors, i.e, has SLOPE. If you hit a dune you can drift laterally. If you hit one fast enough, you will fly through the air for a small distance before your wheels touch terrain again.

3) MUD PATCHES: A mud patch is a blob of mud. While on mud, you have very little traction (low coeff. of friction), so it is easy to fishtail, and hard to change direction if you are going fast. (It is like ice patches in other driving games). On mud, it is best to be easy on the gas and easy on the steering wheel. Either gear can be used, depending on speed.

4) GRAVEL PATCHES: A gravel patch is an expanse of finely distributed, small rocks. Driving on a gravel patch causes you to skip and drift slightly. Small, random variations will be imposed on your direction, and on your lateral position. Note that this is not like what happens on mud, which is predictable. Driving on gravel involves some RANDOMNESS of motion. You need POWER to cross a gravel patch. You can go the fastest if you are in LOW GEAR. Driving in high gear tends to burdon the engine, which will make an uneven, "painful" sound.

NOTE: It is possible to bypass a dune, mud patch, or gravel patch.

5) WATER: See Fig. D. Water appears as rivers. A river is approximately perpendicular to your direction of motion, so that you must cross a river. You can't bypass it. Going through water slows you down very quickly. Your speed drops off linearly and sharply, until it hits some minimum speed of, say, one eighth maximum speed. This linear decrease means that the best way to go through water is to hit it with as fast an initial speed as possible, and to cross a river at the narrowest section possible. Once your speed has dropped to the minimum, your engine will make a gurgling sound. Water is really just another surface, with the additional property that it slows you down, regardless of what you do with your controls. You have esentially no traction in water. You get across it with pure momentum, until your speed hits the minimum.

As you barrel into water, you "part" it, which will be graphically represented by furrows of water being thrown to the sides. This leaves a waterless gap in the river. The surface of this gap is river bottom, which will have a fairly high coeff. of friction. A gap will remain open for only

about 1.5 seconds, and then the water will rush back into the gap, and water will again exist at that spot. This can be taken advantage of: since it will often be the case that you are right behind a vehicle, you can let him do the parting of the water, and shoot across the river through the temporary gap he creates. If you see a river ahead, you slow down and put some distance between you and the vehicle ahead of you. This is so you don't rear-end him when he hits the water. You then shoot over the gap, and pass him on the other side of the river, where he will have slowed way down for a second, and will be easy to pass.

Water in a river flows left or right. Particularly wide rivers will cause vehicles to be swept down stream, if they enter the river at a slow speed. As with river gaps, a river's flow can be either good or bad, depending on whether the player can take advantage of it. SEE FIG. D: If a player needs a Nitro Fuel cannnister, the only way he can get the one shown is crossing the river. If he enters the water via Path A, he can let the flow sweep him to the right, and reach the opposite bank at the point where the cannister is located. This carefully timed, slow crossing may be the only way to get at the cannister, considering the location of the rock, and of the dune. A gap created in a river with a flow will skew appropriately.

There is a better way to cross a river: JUMP OVER IT by launching yourself off of a "ramp" dune which runs up to a bank. In Fig. D, we see a river ahead. The river varies in width. If he keeps going straight ahead, he will plow into a narrow stretch of water (Shown as Path A). If he turns right as fast as he can, he will be able to cross the river via the ramp dune (Path B).

Since the terrain is scrolling down pretty fast, there isn't much time to react to new things as they appear at the top of the screen. For this reason, if a ramp dune is going to be usable, it must be at least partially IN VIEW AS SOON AS THE RIVER IS. See Fig. D: the player is able to see a piece of the dune to the right. This is the "worst case" for a river/dune combination: the dune is FAR to one side. Sometimes, the dune will be attached to the river more near the center of the screen.

Since terrain features are created DYNAMICALLY, a river doesn't really go anywhere. Note that the river in Fig. D is truncated at the ends. This is so a stock piece of river can be put up ahead of the player. (Stock pieces of data must be bounded by reasonably small rectangles). There will be several stock pieces of river. The "bar bell" shape shown is narrow in the center and wide at the sides. Other stock shapes will be: wide in the center and narrow at one or both sides, narrow everywhere, wide everywhere, etc. For each of these stock thickness profiles, there will be versions that are angled, in addition to the perpendicular versions. There will also be several stock ramp dunes: narrow, wide, steep, etc. Considering all the combinations of river shapes with ramp dunes, the player shouldn't percieve the "stockness" of it all. He should simply feel like there are frequent rivers, which vary in width, direction, and jumpability.

Jumping a river via a ramp dune should be interesting; i.e, some ways are better than others. See Fig. E. Look at approach A. Here the player approaches the dune head on. He pops over the river, travelling in a straight path. He lands on the other side stably, and can continue. Now look at approach B: the player has had to angle to the left in order to make

the dune. This means that he flies over the river at an angle. When he lands, the direction he is pointed in is different than the direction he is travelling in (with respect to the terrain). If the difference between these angles is great enough, he will not land stably: he lands, bounces, and ROLLS a few times, coming to a stop, in the upright position. He then may accelerate back up to speed. Here, the penalty is loss of time, as always.

If you go off a ramp dune at a slow enough speed, you will not make it all the way to the opposite bank. You will plop into the river, and have to traverse the remaining water at minimum speed. You will at least have flown over some of the water, so landing short is better than not using a dune at all.

As with rivers, the other terrain types: dunes, mud patches, and gravel patches, exist in stock variations. There will be as many sizes and shapes of dunes, and of patches, as necessary.

Terrain is generated dynamically. This means that the frequency, order, and relative distribution of the various surfaces can be controlled and tweeked. This allows for interesting terrain combinations: consider a ramp dune with a mud patch just before it, or a long gravel patch just before a river bank (the gravel would slow you down, when you want to hit the water fast), or a consecutive series of rivers, with ramp dunes lined up in a straight path: you can HOP RYTHMICALLY over river after river.

$$$

****
P A S S I N G
****

Opponent vehicles will try to stay in front of the player, making passing a challenge (See Technical Notes for more detail on implementation).

SEE FIG. F. The player wants to pass vehicle X. X is very adept at hogging the road, so the player uses a dune. When the player sees the dune scrolling down, he snaps the wheel to the right, in order to bypass the dune. X snaps to the right along with him, in an attempt to stay in front of the player. X will hit the dune, while the player does not. This causes X to be deflected to the left by the dune, and the player passes him on the right. The player has used the dune as a "wedge" to seperate himself from X.

Another example of passing is shown in Fig. D: If the player takes Path B, X will zoom to the right as well, right into the river!

The player can use this method to lure the vehicle ahead of him, or the one BEHIND him, into any kind of thing: mud patch, gravel patch, rock, cactus, etc.

Ramming sideways into a car can also help in passing it. If you get the front of your car ahead of an opponent car's rear, you can bump him to the side, then quickly slip by while you and he are rebounding away from each other. Ramming too hard causes you to roll out.

****
THE PLAYER'S OPPONENTS
****

Now that the environment and rules of motion have been established, we can easily define various opponent vehicles and their characteristics. SEE FIGURE C:

1) Generic, ordinary truck. He is like your vehicle. He has average capability, and reacts to terrain about like you do.

2) ROAD HOG. He actually looks like a hog. He is particularly good at staying right in front of you.

3) MUD MASTER. He has a small body and large balloon tires (like "Big Foot"). He is very good in the mud (i.e, he is less susceptable to a low coeff. of friction). He kicks mud back in your face (See Screen 2), which slows you down after a while. You can wash off the mud he has piled up on your vehicle by driving through a river. The effect of too much built-up mud on your vehicle is less responsive handling due to top heavy weight distribution.

4) ROCKY. He is good going over rocks, and gravel fields. He has special tooth-like treads on his tires. He kicks rocks back at you, which tend to slow you down.

(Note: You can also kick back mud and gravel, doing to them what they can do to you).

5) V-DUD. He is a beat up old VW, circa 1960's. He is held together with bailing wire, bandages, etc. His only talent is that he looses parts of himself, especially over rough terrain. The parts he leaves behind are like rocks if you hit one; they jolt you in another direction, and knock down your speed.
An alternative to this guy could be an old truck, piled high with all of the driver's worldly goods: furniture, TV, matresses, tools, etc, which he similarly looses. He would look like the Beverley Hillbillies' truck when they "packed up the kids, and they moved to Beverley".

6) ROCK IT. A jet powered vehicle. He is nearly immune to terrain. He just blasts his way through anything. He is a tough customer. If you get close behind him, you get tourched by his exhaust, which causes you to heat up to a cherry red, and to lose some engine power. You will cool off slowly, unless you drive through water, where you will quickly be revived (lots of nice steam and hissing will occur). It may be the case that the only way to get past him is to use your "jet boost" button, as described in the Controls section.

With a dynamic terrain generator, we can easily make things interesting with regard to those opponent vehicles which have a favorite type of

terrain. For example, consider the Mud Master. You are faster in regular sand than he is, but he is faster in the mud than you are. When you are trying to pass him, the generator could deliberately be biased toward generating mud patches. This means that you must deal with him "on his turf". To pass him, you will have to weave between the mud patches, trying to stay on sand as much as possible. He will not be much of a road hog, since he already has this advantage. The same thing will happen during confrontations with Rocky, except with gravel patches.

There are no doubt other interesting personalities that could be in the game.

****
G O O D I E S
****

A "goodie" is a useful object of some kind that sits on the terrain, and can be picked up by the player if desired. Goodie locations are shown on the global map. The goodie which has already been discussed is the Nitro Fuel Cannister. Note, when the player goes after a goodie, he never knows what kind of terrain he will find it on. He may navigate himself to a nice can of Nitro Fuel, but when he gets there, he discovers that it is in the middle of a river! It may still be advantageous to pick it up, even though he has to go through water to do it.

Another example of a goodie is a tire. There are spare tires located at various places. The wear on the player's tires will be shown graphically (4 levels: new, worn, bald, and rim only), and if the player wears down his tires enough, his ability to turn will start to degrade. At this point, he would want to go pick up new tires, if he can find them.

Tire(s) which hit a cactus will immediately drop down 1 level of wear. Whenever a tire advances to the last (rim only) wear level, a bang is heard and the final strip of rubber is shown blowing off of the rim.

Since the player's vehicle is essentially a pickup, he can collect goodies for future use. When a tire reaches the final stage of wear, it will automatically be replaced by one he has previously picked up.

****
O B S T A C L E S
****

An obstacle is an object, other than an opponent vehicle, which hinders you in some way. They are created dynamically by the terrain generator, so they are not on the global map. They just pop up at inconvenient times:

1) ROCKS: Small and large rocks are strewn about on the terrain. Note, a rock is different than the components of a gravel field, which are really little rocks, but grouped together in a distinct patch. Hitting a rock causes a sudden jolt to your progress. Hitting a small/large rock causes your direction to change by a small/large amount, and your speed to immediately drop by a small/large amount, respectively.

2) CACTUS: Causes a jolt like a big rock, and causes immediate wear on the tire(s) which smash the plant.

3) TUMBLE WEED: A ball of tumble weed will occasionallly sweep toward you laterally. If you drive over one, it gets stuck under your front wheels, for a short while, until it finally gets crushed, and dissipates. While it is stuck under your vehicle, YOU CANNOT TURN YOUR FRONT TIRES; your steering is temporarily frozen.

****
C A N Y O N S
****

SEE FIG. G. A canyon is a cross between a terrain type, and an obstacle. A canyon sneaks up on the player. He cannot circumnavigate a canyon, because its entrance is a wide, gradually narrowing, mouth. A canyon is a winding rut. It is bounded by two vertical walls. In Fig. G, the bounding walls are red. The perspective is fake projective, as in Equitus (the left wall goes down and to the right, the right wall goes down and to the left).

The bottom of the canyon is shown as cross-hatched. A canyon is effectively a temporary ROAD. It eventually dissipates. For the time a player is in a canyon, he must follow its path as if it were a road. If he hits a wall, he will slow down. If he hits it hard enough, he will roll and stop, as with a big rock. The other vehicles are also forced to go through the canyon. The other vehicles will go through it in single file. It is possible to pass a vehicle in a canyon, but this will be tricky, since the other vehicles will be going close to the maximum possible speed, without fishtailing. The other vehicles will simply follow the path of the canyon, all at the same speed.

It is possible to exit a canyon at certain shallow areas. Note the stretches marked as "X" in Fig. G. The player can break out of the canyon by passing over these parts of the canyon's boundary. The player cannot go through a canyon SLOWLY, so that he can hunt for an early exit, and then take it safely. Going too slowly will cause the vehicle behind him to rear end him.

****
T E C H N I C A L N O T E S
****

Normally, your vehicle will be driving on terrain surface. (The exceptions are when it drives over obstacles on the terrain, like rocks, cacti, etc, discussed later). The terrain surface has TWO properties:

1) Static Coefficient of Friction.
2) Normal Vector.

If the torque of your vehicle's tires exceeds a threshhold, traction is "broken" and slippage begins. The threshhold equals the static coefficient of friction times weight. Weight varies as the vehicle floats up and down over dunes. The torque delivered by the tires to the surface is calculated by a torque curve, the RPM, the gear stick position, and the accelerator's amount of depression. A tabled torque curve will be used. Torque is calculated by indexing into the table by RPM, and multiplying by (pedal depression/gear). The forward acceleration delivered to the vehicle by the tires is directly related to this calculated amount (by the vehicle's mass). If the torque exceeds the threshhold, the tires' push drops, and subsequently a simple dynamic coeff. of friction (say, one eigth the stat. value) will be used: ie, the tires break free and spin a lot, kicking back sand, and RPM increases quickly.

The SAME stat. coeff. of friction is used to determine fish tailing. If the steering wheel position is suddenly changed, then, depending on the vehicle's speed, the vehicle will fishtail if the LATERAL torque of the tires against the surface exceeds the threshhold.

The Normal Vector of the surface determines how the SLOPE of the surface will cause the vehicle to drift left or right, as it goes up the bank of a dune in a direction other than straight uphill. SEE FIG F: Vehicle "X" is swerving to the left, based on the gradient, which is related to the normal.

SEE FIG H. Terrain is divided into SIMPLE areas that correspond to friction and slope. Each dashed region has two values associated with it: Stat. Coeff. of Friction, and a Normal Vector. Regular sand (i.e, default background surface) has a HIGH c. of friction, and a vertical normal. Regions that form dunes have a MEDIUM c. of friction, and normal vectors that reflect the orientation of the dune's surface. Regions that correspond to mud flats have a LOW c. of friction, and vertical normals.

The above mathematical model is tentative and incomplete, and will be hammered out until driving feels right.

To insure that the other vehicles enter the screen's window as you pass them, or as they pass you, a "confrontation hour glass" exists around the player, like an invisible field: SEE FIG. I. This shape causes a compression of lateral position: the vehicle ahead of the player must stay inside this hour glass boundary. This means that passing is done with slim margins: the vehicle you are overtaking is squeezed through the neck of the hour glass, coming close to you. If you turn left or right, the hour glass field moves

with you. This means that the vehicle ahead of you will be dragged along. The effect of this hour glass containment will be that vehicles ahead of you will tend to stay directly in front of you, i.e, IT WILL LOOK AS IF THEY DON'T WANT TO BE PASSED.

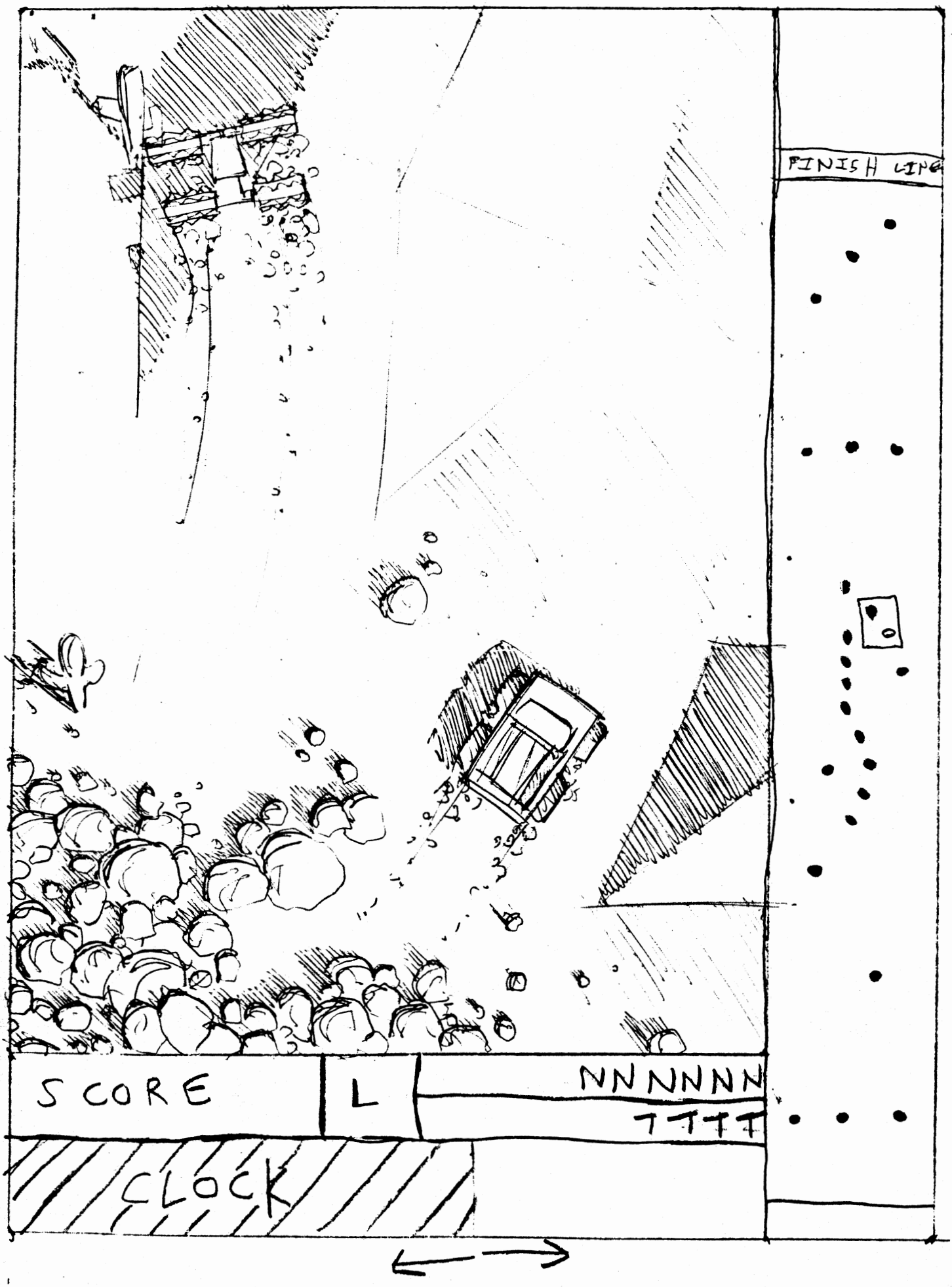
The way a vehicle will obey the hour glass rule will be on the order of: the closer he is to an edge, the more he will turn away from that edge. This means that he is constantly honing in towards some location in front of you. Different vehicles will hone in with different degrees of swiftness.

From Fig. I, note that an hour glass boundary can be described by the triplet (theta,H,W).

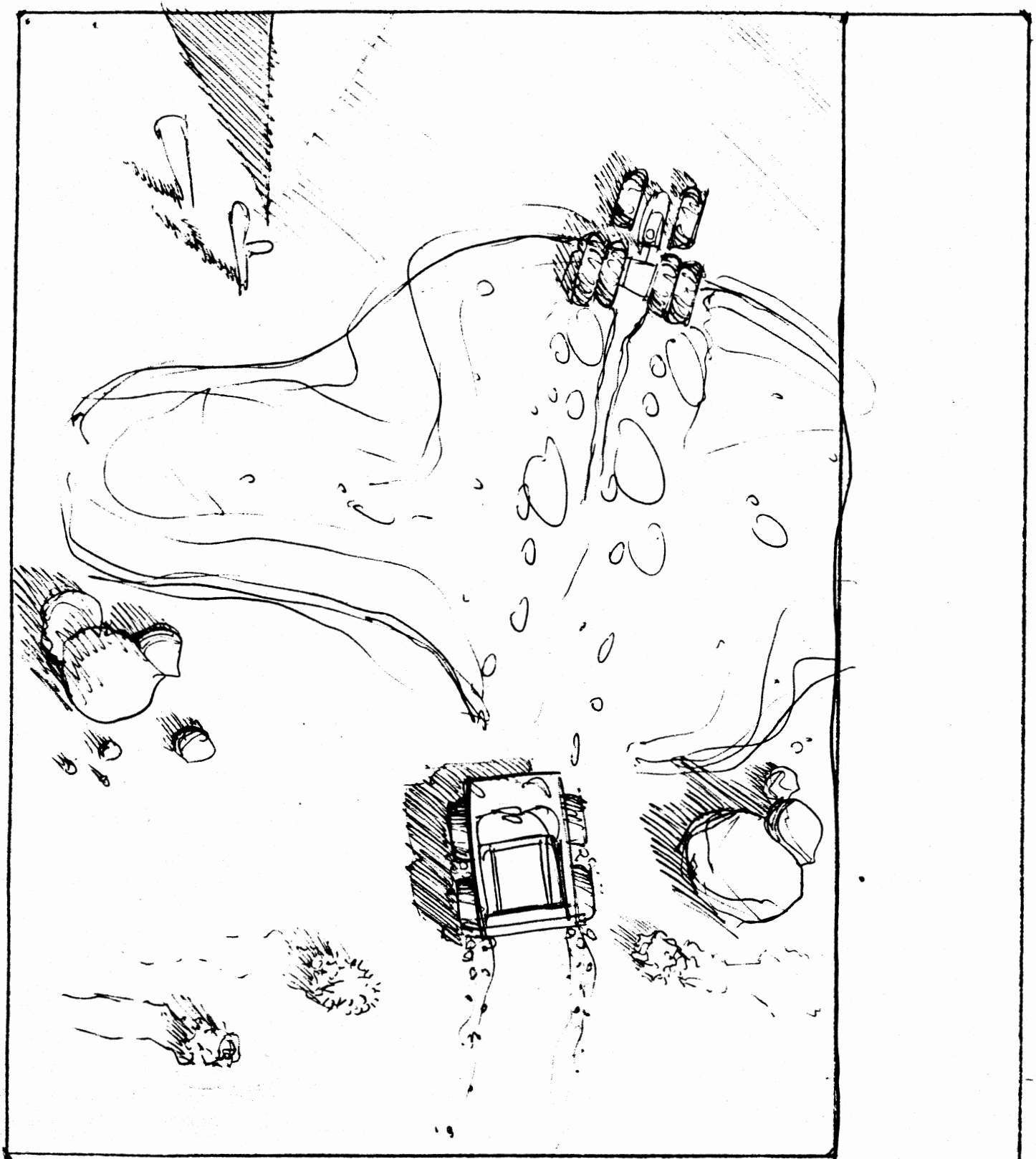
Fig. I shows the player passing vehicle "X" on the left. X doesn't hone very well, he just hugs the right edge of the hour glass. Also, W is large enough, so passing is a simple matter of driving to the left, and letting X slip by on the right.

In Fig. I, note the "rule boundary". This is the area where terrain is meaningful. Terrain doesn't really exist outside of this limit, since it is dynamically generated in front of the player, and it goes away after it scrolls behind him. Other vehicles that are far ahead or behind the player are outside this boundary. They simply exist as a location on the global map. They don't have to obey the terrain, since there isn't any where they are. As you get close to another vehicle, he enters the rule boundary. He must then obey the terrain, i.e, react to a dune, slow down if he passes over gravel, slow way down if he hits water, etc. He must be affected by these things as you are, or it will be unfair. The rule boundary is larger the screen, so that a vehicle has time to "get into synch" with the terrain, before entering the visible screen window.

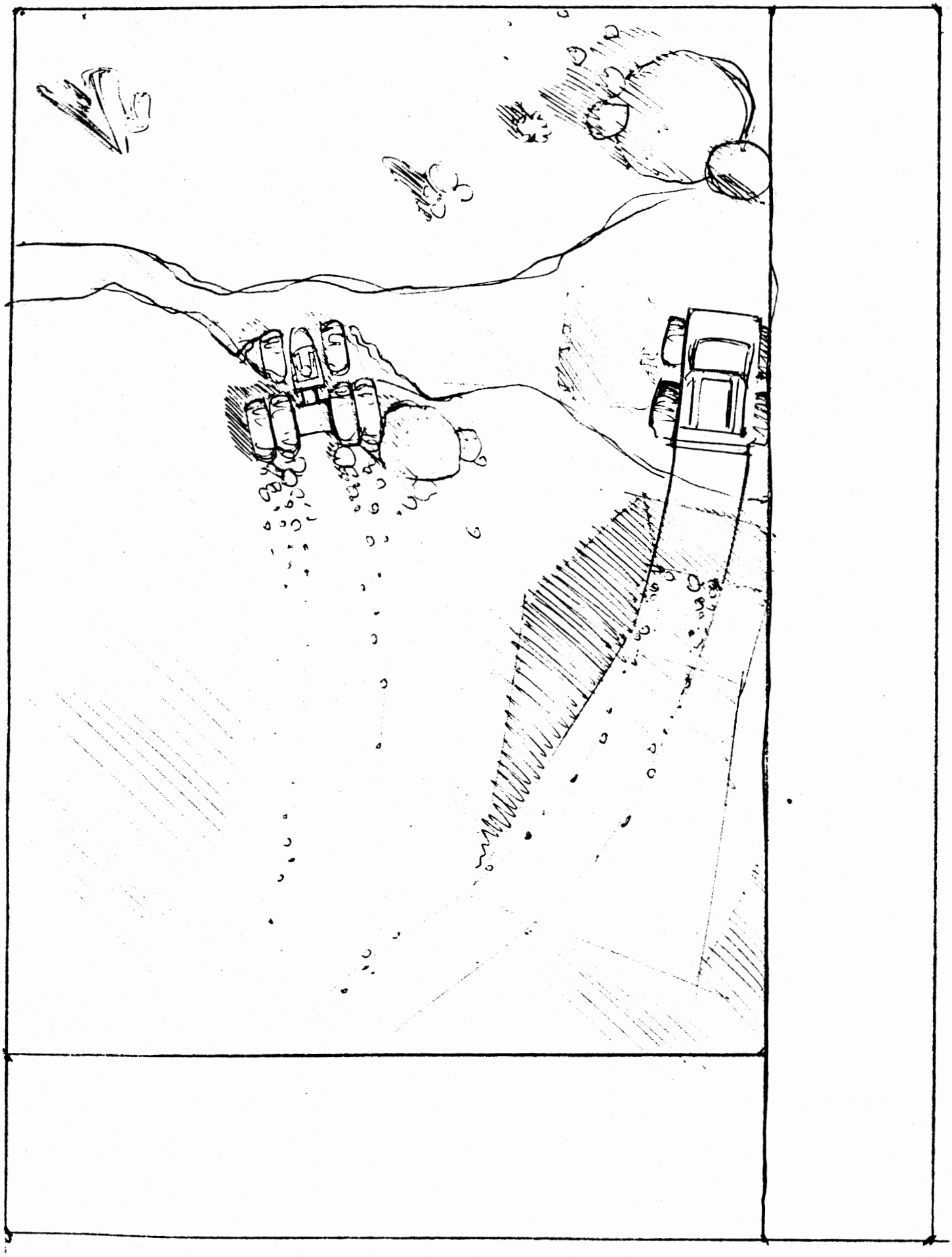
The two sets of rules that an opponent vehicle must obey: the hour glass and the terrain, could conflict. For example, if he hits water, he will slow down, and cant keep up with the hour glass edge. This is the priority: first obey the terrain, and only after that is satisfied, obey the hour glass.



SCREEN 1



SCREEN 2



SCREEN 3

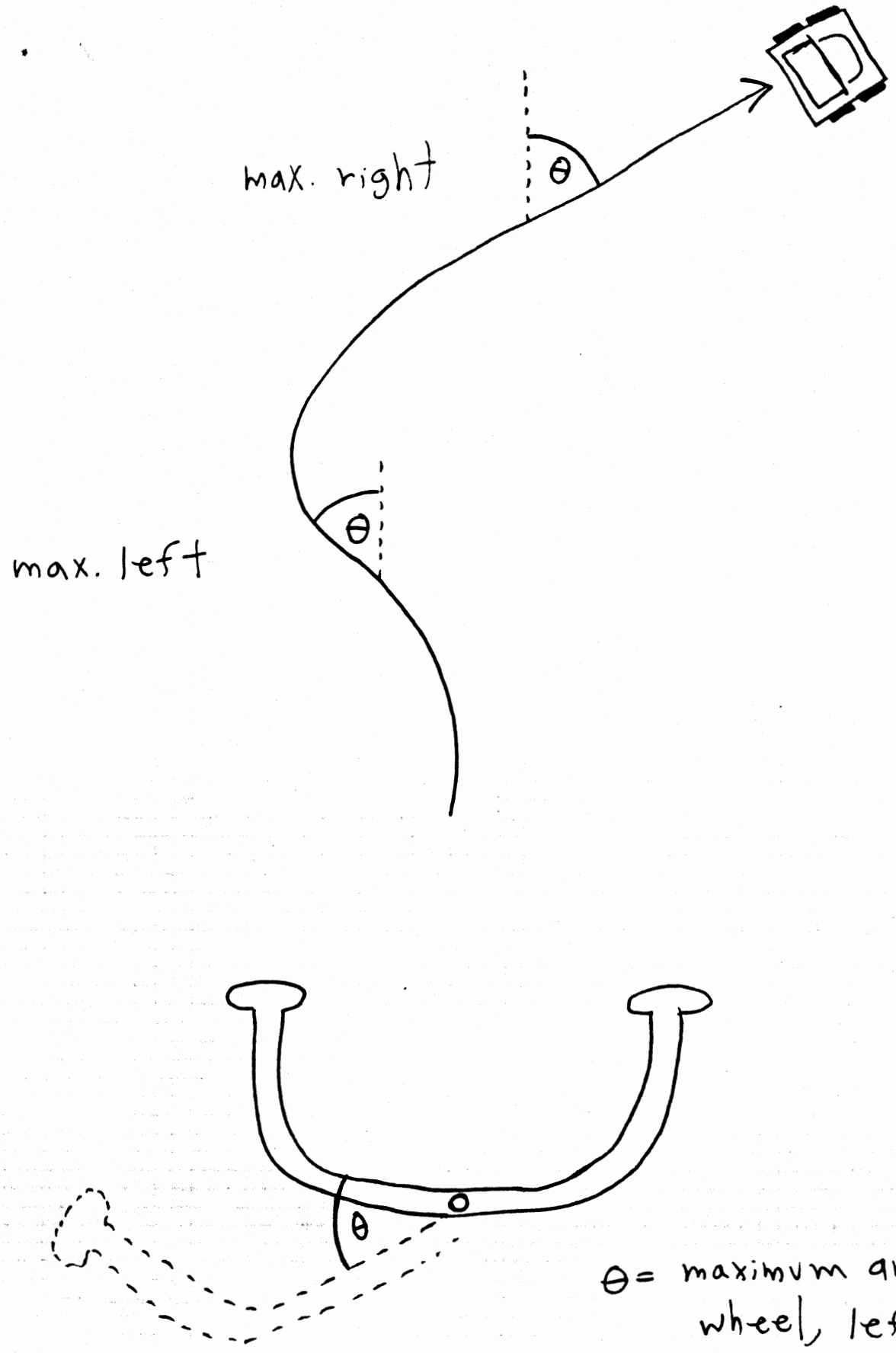


FIGURE A. Turning

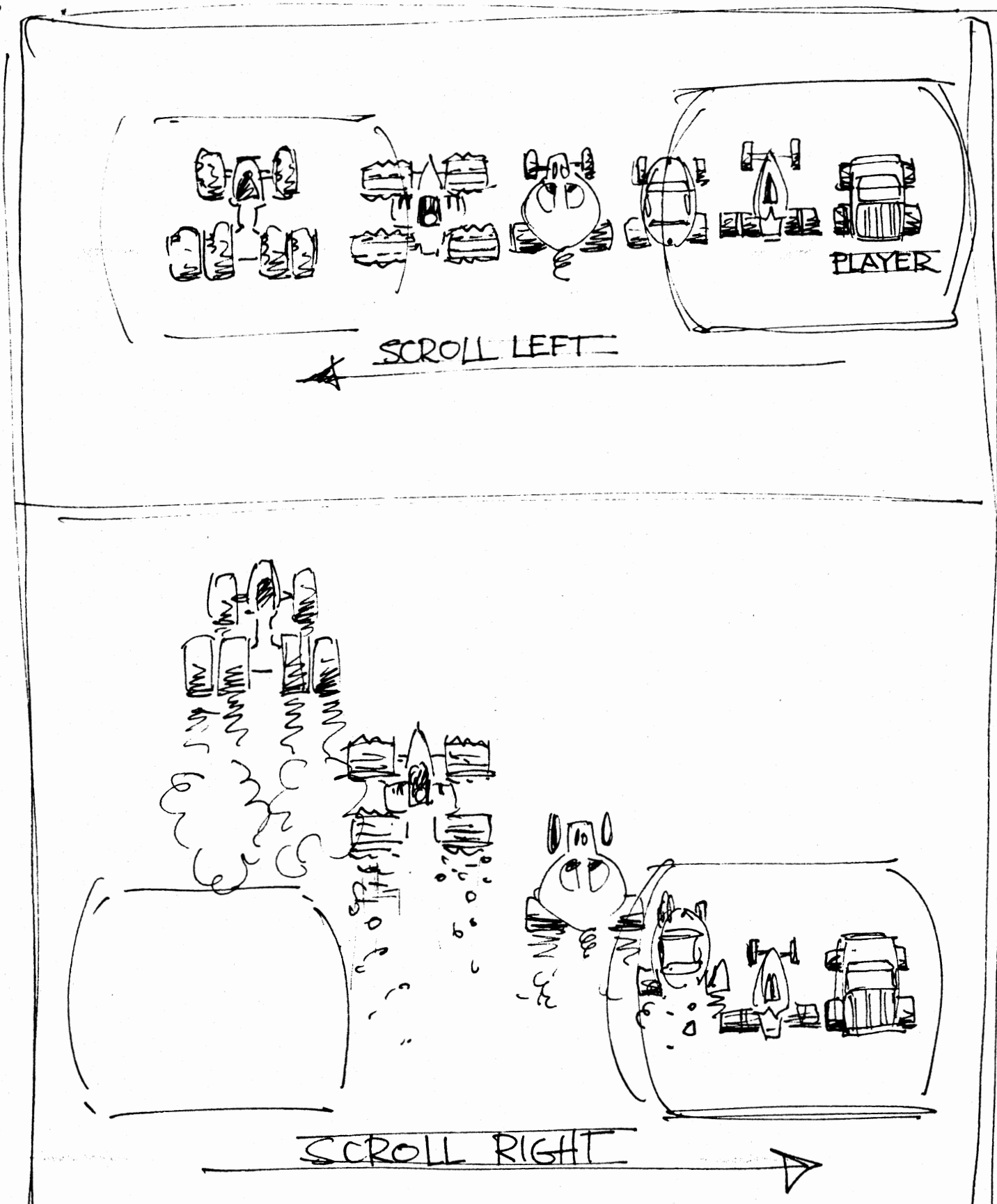
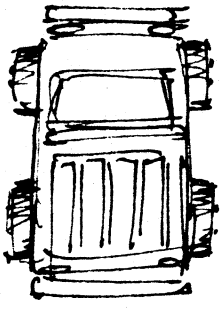
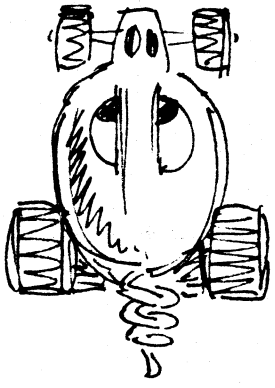


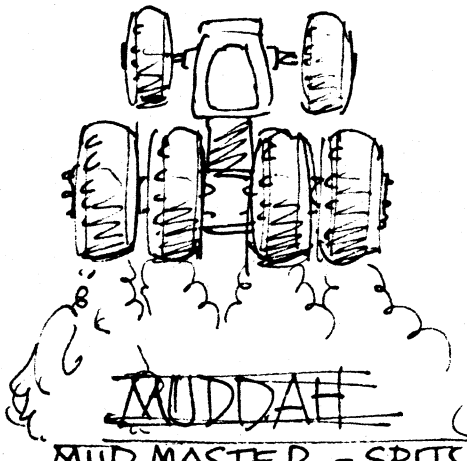
FIGURE (B) START OF RACE



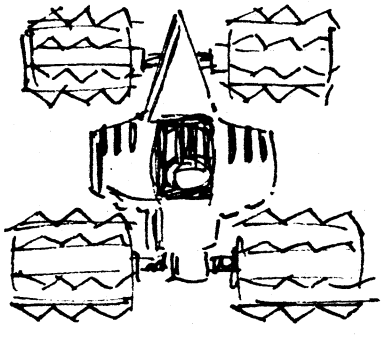
GENERIC
TRUCK
THE PLAYER



ROAD HOG
OINK . . . SQUEEL

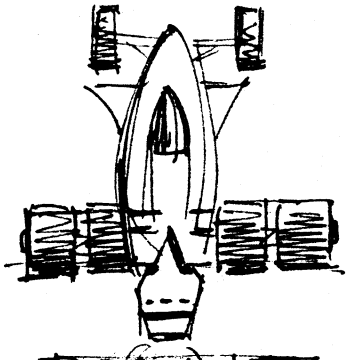


MUDDAH
MUD MASTER - SPITS
ROCKS AND MUD WITH BIG
BALLOON TIRES



ROCKY
SPECIAL TREADS
FOR GRAVEL

V-DUD
DROPS ENGINE PARTS



ROCK IT
SHOOTS BACK FIRE
FROM JET ENGINE

FIGURE (C) YOUR OPPONENTS

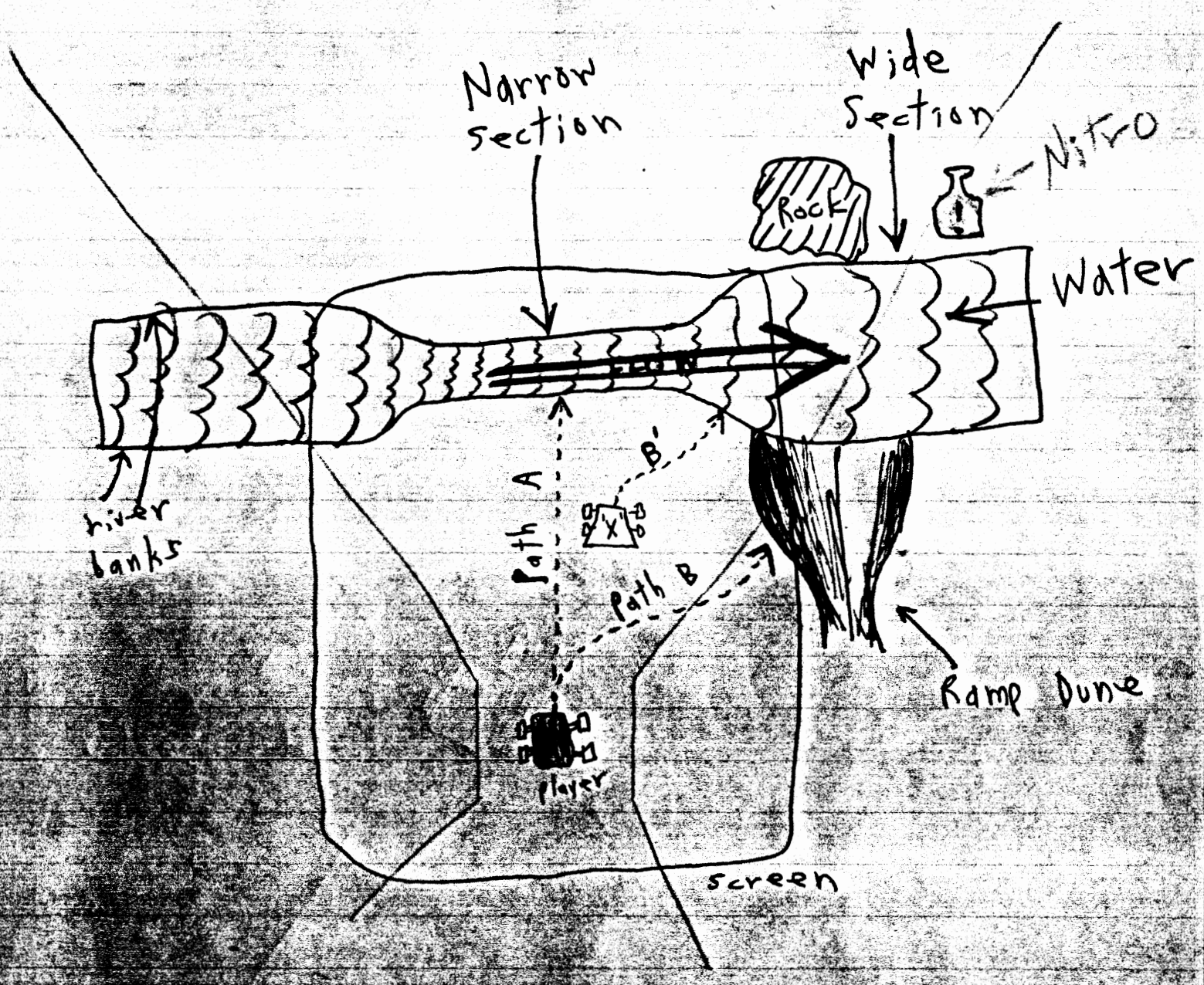


FIGURE D. Rivers: crossing water

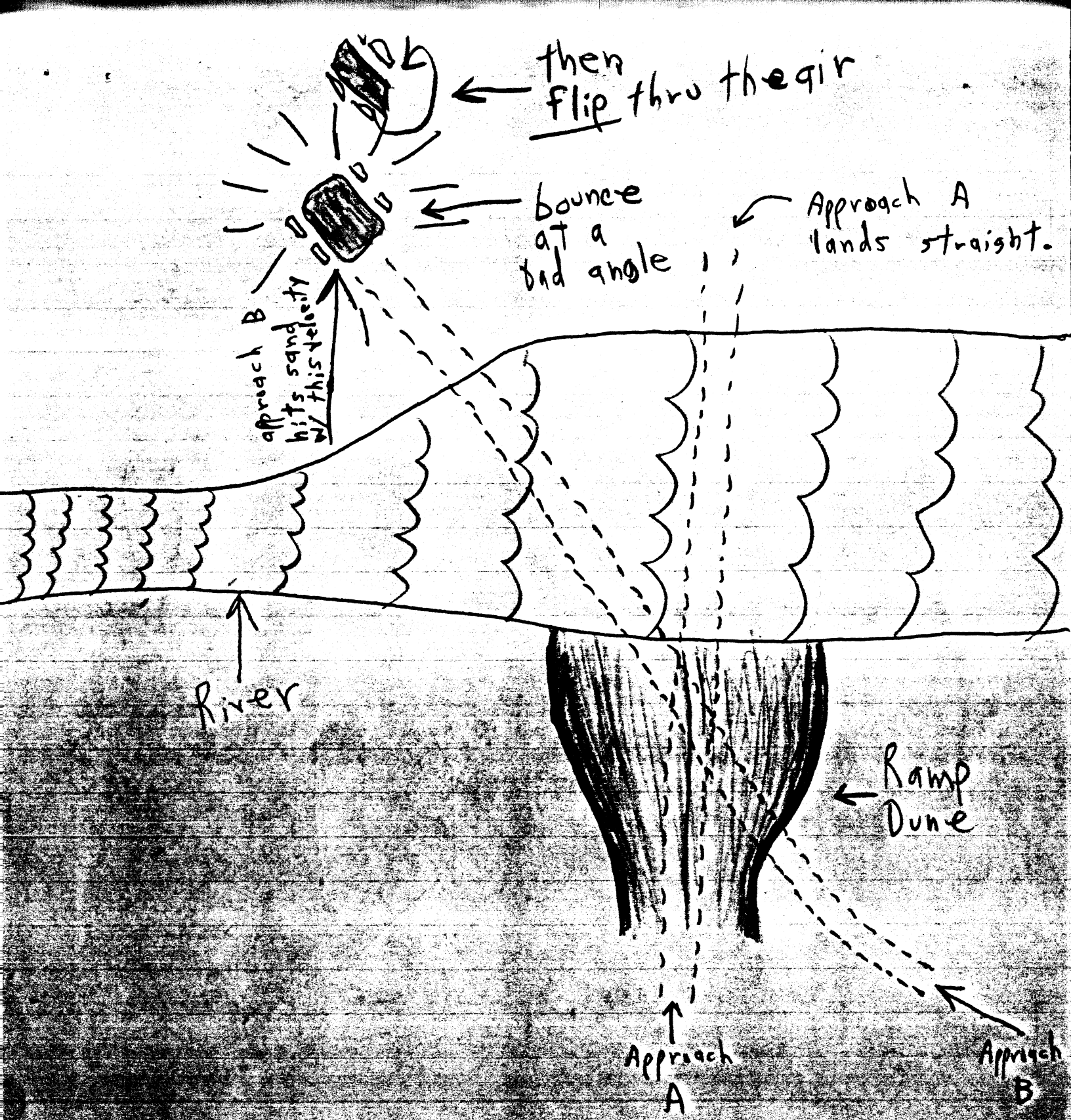


FIGURE E. Hopping over a river

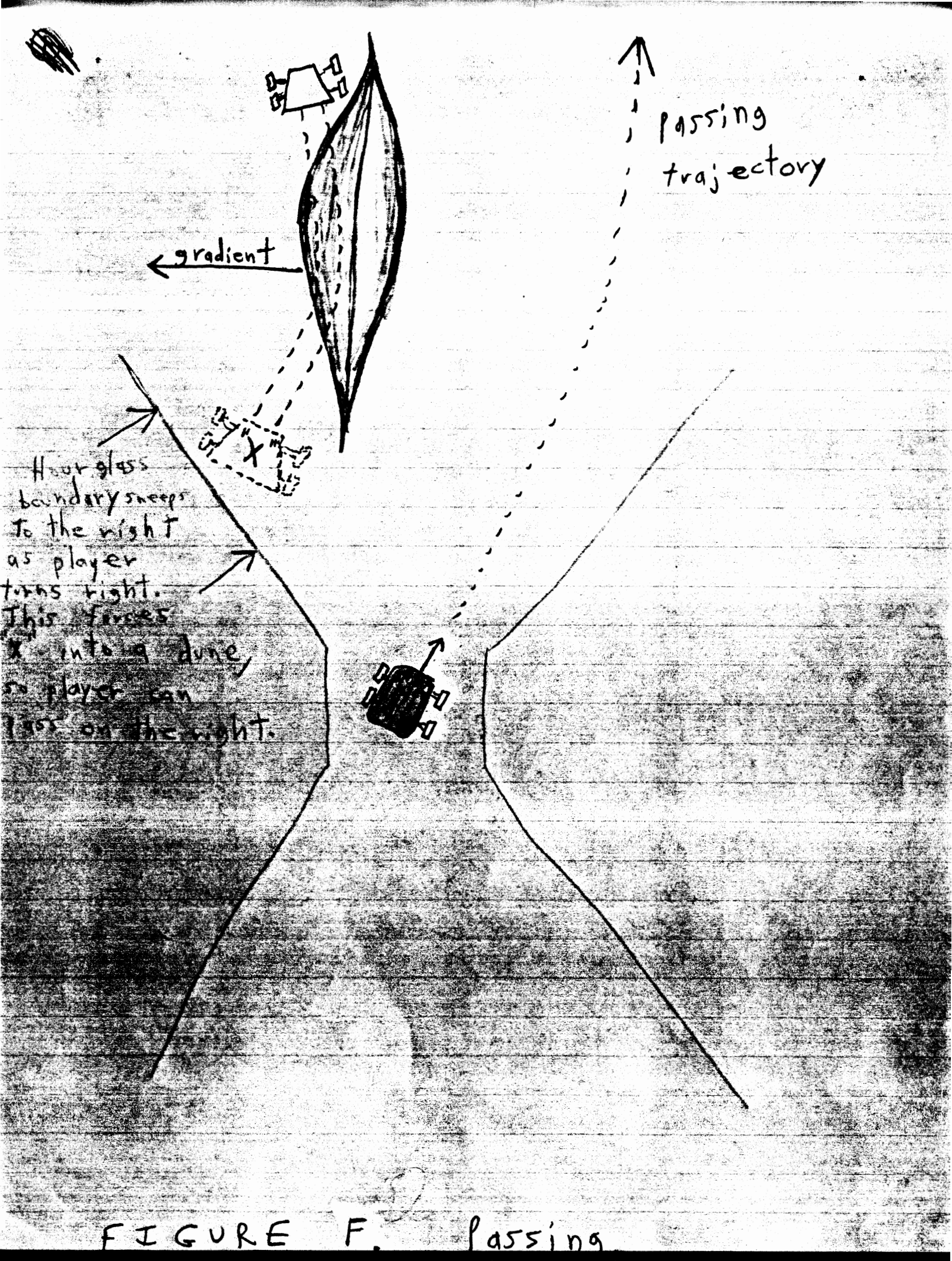


FIGURE F. Passing

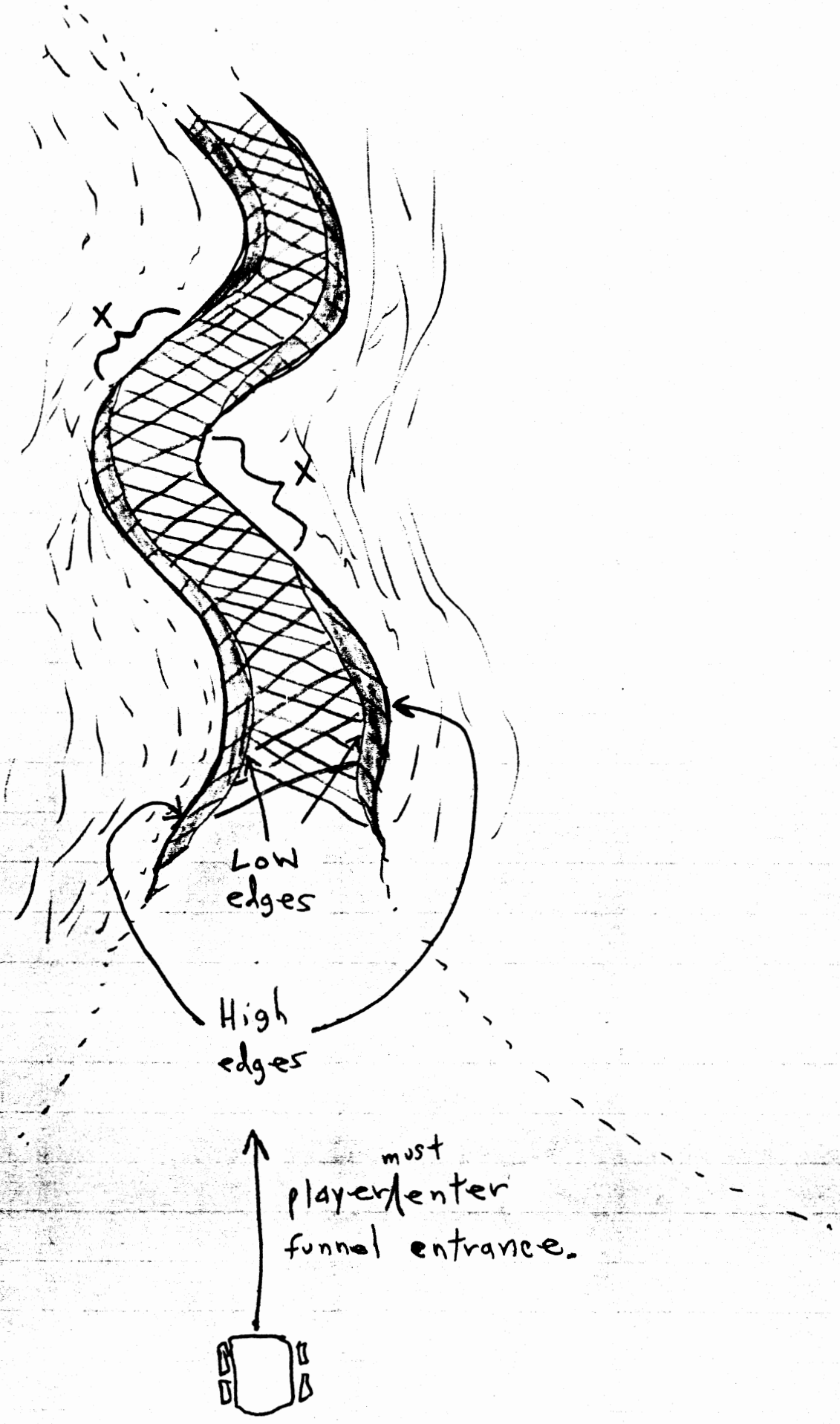


FIGURE 6. Canyon "Road"

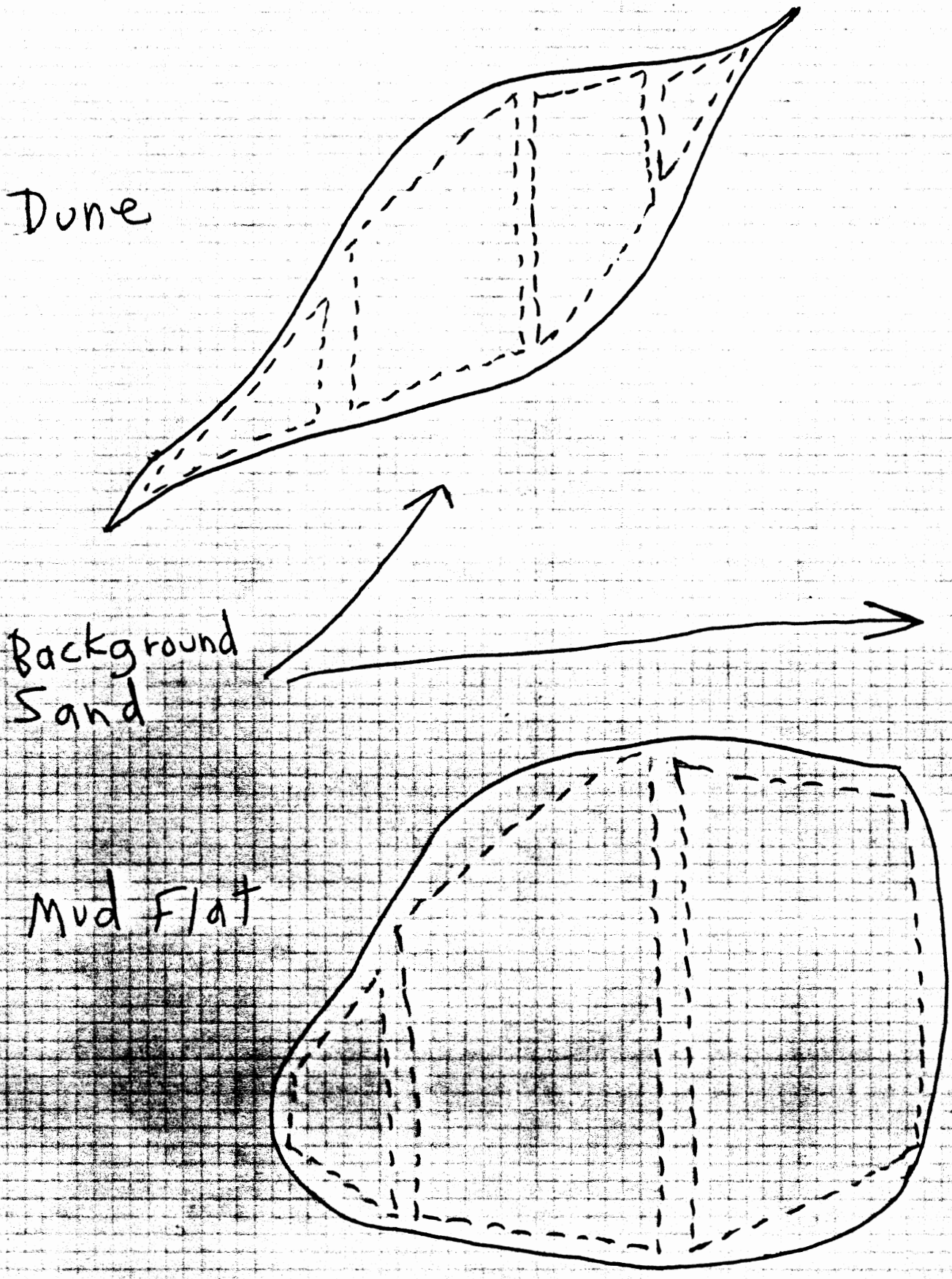


FIGURE H. Logical Division of Surface

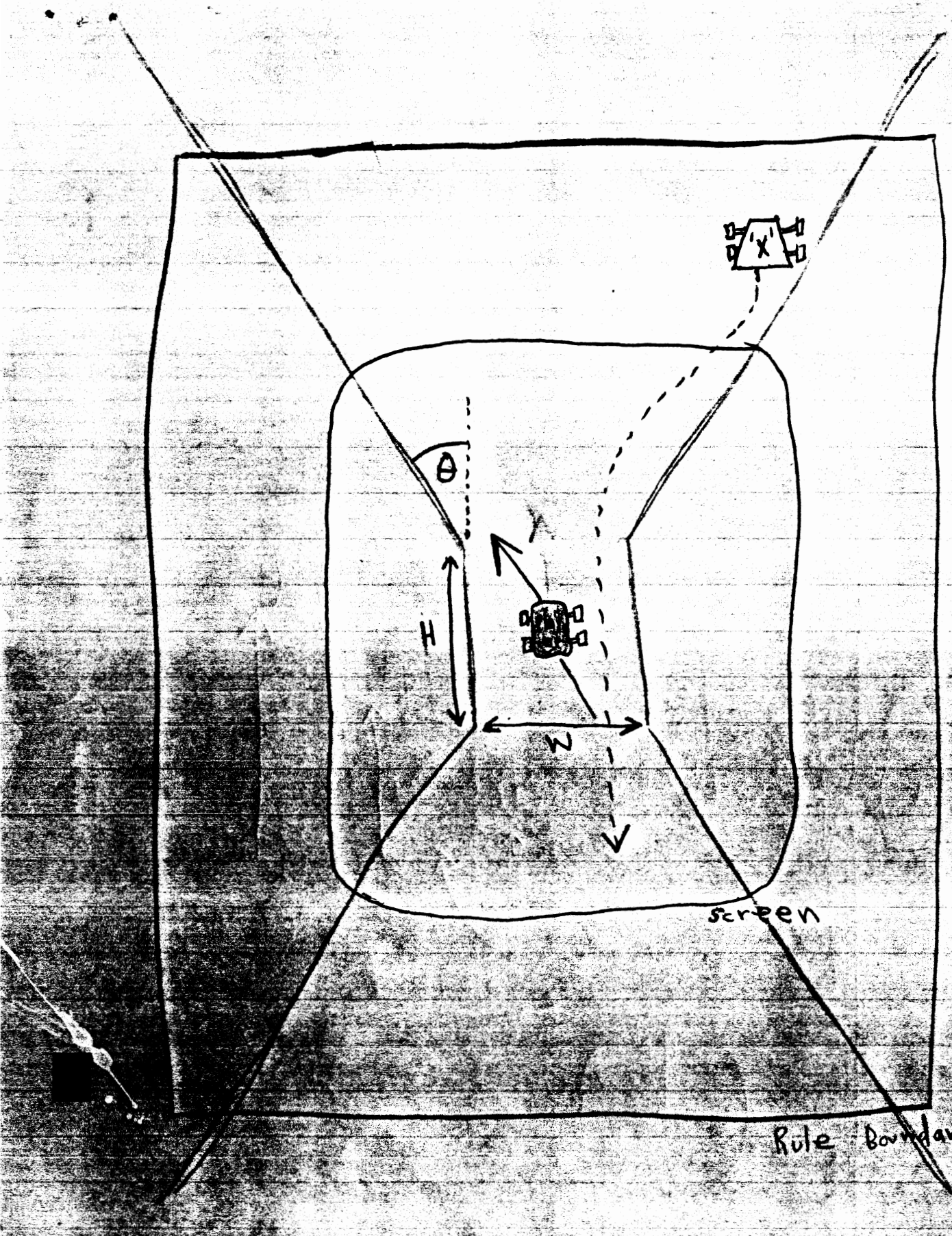


FIGURE I. Confrontation Hourglass: (θ, H, W)

GAME DEVELOPMENT STATUS: OFF ROAD

DATE: 1 February 1985 PROJECT LEADER: Moore PROJECT START: Feb., 1985
LAB: Minh's EXT: 2841 PROJECT NUMBER:

SCHEDULE OR (OCCURANCE)

Project Start	February, 1985
1st Engineering Rev.	9 April 1985
1st Marketing Rev.	May, 1985
2nd Engineering Rev.	June, 1985
2nd Marketing Rev.	July, 1985
Focus Group	August, 1985

Project Schedule:

February, 1985	Project Initiation/Start.
April 9, 1985	First Engineering Review. Show player cars interaction with terrain according to traction and torque. Show simple version of Terrain Generator. At this point, the player can drive his car from start to the finish line, with some sliding and bouncing. No complex terrain interactions. One type of opponent car: generic. Show Global Map.

HARDWARE DESCRIPTION:

Processor: 68010 @ 7.16 mHz
Program RAM: 2K Words

Graphics: System I - vertical monitor
Display: 240 X 336 Pixels (raster)
Motion Objects: Up to 56 8 X 8n high, + additional MO's for Map
Playfield: 30 X 42 stamps. A stamp is 8 X 8 pixels.
M.O. and P.F. stamps are 4 bits deep.

PROJECT TEAM:

Project Leader: Rich Moore
Programmer: Jack Ritter
Engineer: System I
Technician: Minh van Nguyen
Graphics Consultant: John Bell
Sound/Speech Support: Brad Fuller

DELTA COST COMPARISON FOR "OFF ROAD" KIT

EPROM			ROM		
OFFROAD	MARBLE	DELTA (1)	OFFROAD	MARBLE	DELTA
1) GRAPHICS MEMORY (2)					
12 27128-3's (3)	13 27128-3's		4 23256-3's 4 23128-3's	5 23256-3's 3 23128-3's	
		$-8.50			$-2.50
2) 68010 MEMORY					
10 27128-2's	10 27128-2's		4 23256-2's 2 23128-2's	4 23256-2's 2 23128-2's	
3) 6502 MEMORY					
2 27128-3's	2 27128-3's		2 23128-3's	2 23128-s's	
4) CART (PCB, Slapstick, map, pal, etc.)					
$ 47.79					
TOTAL CART COST					
$251.79	$260.29	$-8.5	$150.44	$152.94	$-2.50
5) CONTROLS					
(Steering Asm: $41, Shift Asm: $7, Foot Pedal Asm: 30, Turbo Button: $4)					
$ 82.00	$ 65.87	$16	$ 82.00	$ 65.87	$16
GRAND TOTALS					
$360.56	$352.93	$ 7.50	$232.56	$218.93	$13.63

NOTES:
1) DELTA = OFFROAD - MARBLE
2) Assuming $8.50/27128, $3.50/23128, $6.00/23256
3) 3K MO stamps @ 4 bits + 3K PF stamps @ 4 bits = 3 X 2K X 4 = 12 2x128's